Today I will show you how you can easily develop an app for an Android device, mobile phone or tablet with the help of graphic elements, for example to control the robot car. Welcome to the new episode of

Micropython on the Robot Car

Part 3 - Controlling the car by a mobile phone

The mobile phone recognizes the position in the room and automatically changes the screen display, for example, depending on the orientation of the device. Behind this is a relative of the MPU6050 chip, which is built into the GY521 module that we used in our hand control in the first episode. Maybe you tried this feature together with your Robot Car in the second episode, what am I talking about, of course you have already done that. Well, whatever works with the hand control you can do with your cell phone after following the steps in this description. Don't worry, it won't be complicated. If you loved to play with building blocks when you were young, you can easily master it.

Obtaining and installing the software

The hardware is basically complete, unless you want to give your car two LEDs for the taillights. What you need now and here are therefore only two things, the robot car and your Android smartphone. Well, then there's the software. There are three ways to use the free software from MIT (Massachusetts Institute of Technology), which is also a licensee. Two of them lead through software parts that have to be installed on the PC. I chose the third option, which is also highly recommended by the makers of App Inventor 2 (AI2). Only a browser is used on the PC and the MIT AI2 Companion app on the mobile phone. A browser like Firefox or Chrome does it on the PC. It definitely doesn't work with Internet Explorer, but it's dead anyway. I haven't tested whether Edge can be used, what for? You can install the MIT AI2 Companion via the Google Playstore. For this procedure to work correctly, the mobile phone must be logged into a local radio network, not via the Internet. Access via the Internet cannot work because any number of destinations can be reached from the cell phone, but conversely, the cell phone cannot be addressed directly from the Internet. It is encapsulated in the provider's local network and is therefore not visible from the outside. This means that the browser cannot access the mobile phone from the Internet because there is no public IP and no routing.

The basic tool is now ready. However, there is a small handicap: the cell phone can use the TCP dialect perfectly, but at first doesn't understand a word UDP. This is also not necessary for the standard applications. So we give him a little language course via AI2. I found it on Ulli's robot site. Here is the download address:

https://ullisroboterseite.de/android-AI2-UDP/UrsAI2UDP.zip

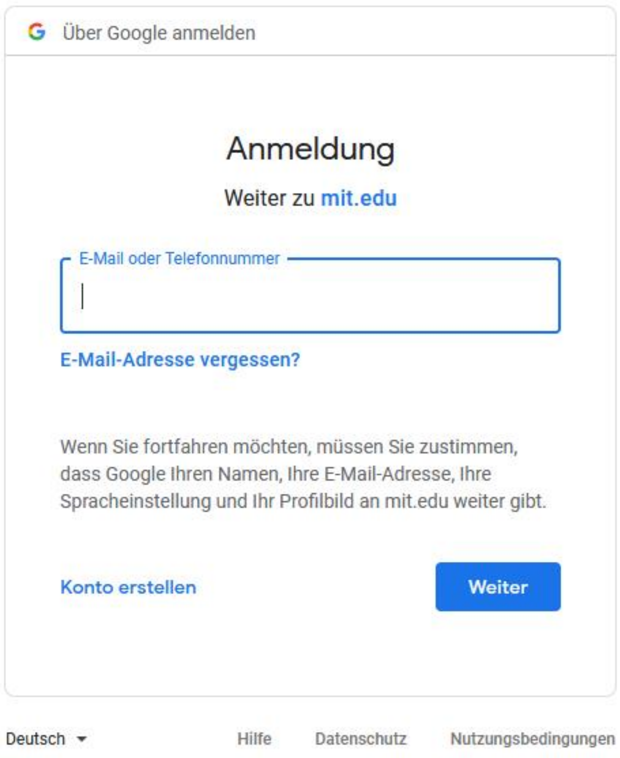
First unzip the archive into a directory of your choice. Go there and make sure that the file de.ullisroboterseite.ursai2udp.aix is listed. It contains the extensions that we will shortly need to create our tax app.

Starting App Inventor 2 for the very first time

Now open Chrome or Firefox, if no browser is running yet, and enter the following URL.

http://ai2.appinventor.mit.edu

You can only register with Appinventor with a Google account. If you already have one, click on Next, otherwise on Create account.


Über Google anmelden

Anmeldung

Weiter zu mit.edu

E-Mail oder Telefonnummer

E-Mail-Adresse vergessen?

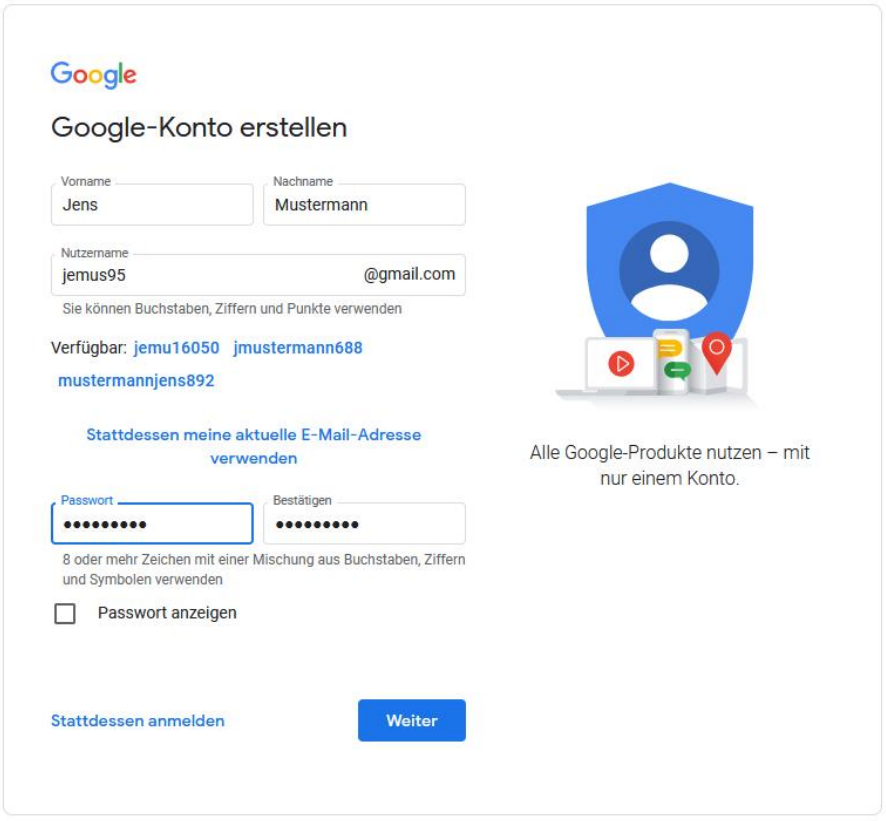
Wenn Sie fortfahren möchten, müssen Sie zustimmen, dass Google Ihren Namen, Ihre E-Mail-Adresse, Ihre Spracheinstellung und Ihr Profilbild an mit.edu weiter gibt.

Konto erstellen

Weiter

Deutsch ▾ Hilfe Datenschutz Nutzungsbedingungen

The account can also be created with fake data. This applies to the first and also to the following page.


Google

Google-Konto erstellen

Vorname: Jens	Nachname: Mustermann
Nutzername: jemus95	@gmail.com

Sie können Buchstaben, Ziffern und Punkte verwenden

Verfügbar: jemu16050 jmustermann688 mustermannjens892

Stattdessen meine aktuelle E-Mail-Adresse verwenden

Passwort: •••••••••	Bestätigen: •••••••••

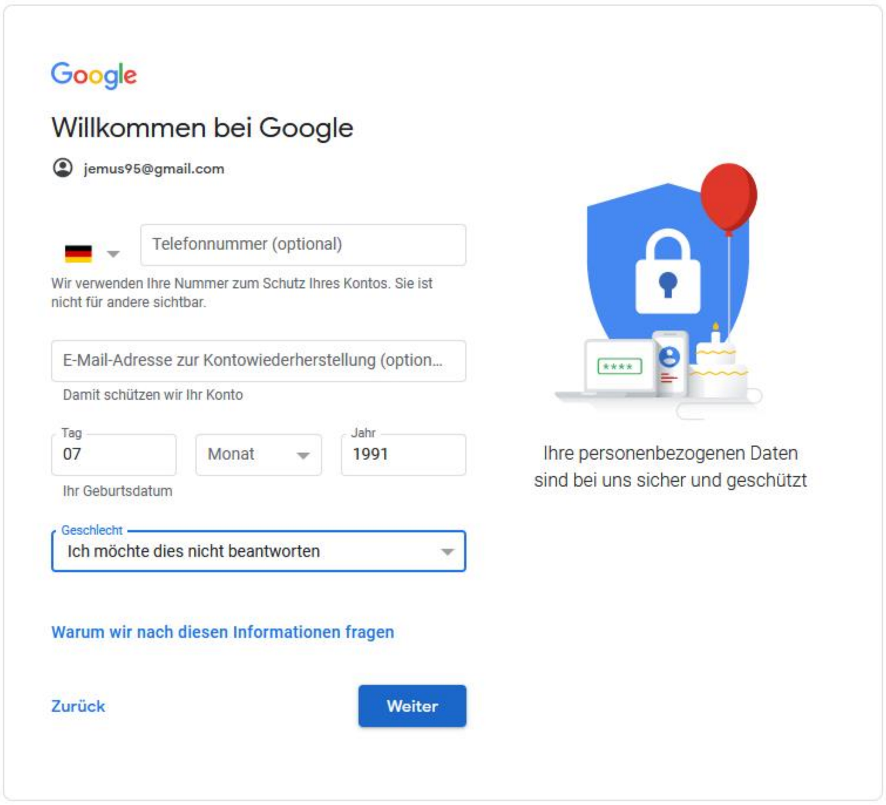
8 oder mehr Zeichen mit einer Mischung aus Buchstaben, Ziffern und Symbolen verwenden

☐ Passwort anzeigen

Stattdessen anmelden

Weiter

Alle Google-Produkte nutzen – mit nur einem Konto.


Google

Willkommen bei Google

jemus95@gmail.com

Telefonnummer (optional)

Wir verwenden Ihre Nummer zum Schutz Ihres Kontos. Sie ist nicht für andere sichtbar.

E-Mail-Adresse zur Kontowiederherstellung (option...

Damit schützen wir Ihr Konto

Tag: 07 | Monat | Jahr: 1991

Ihr Geburtsdatum

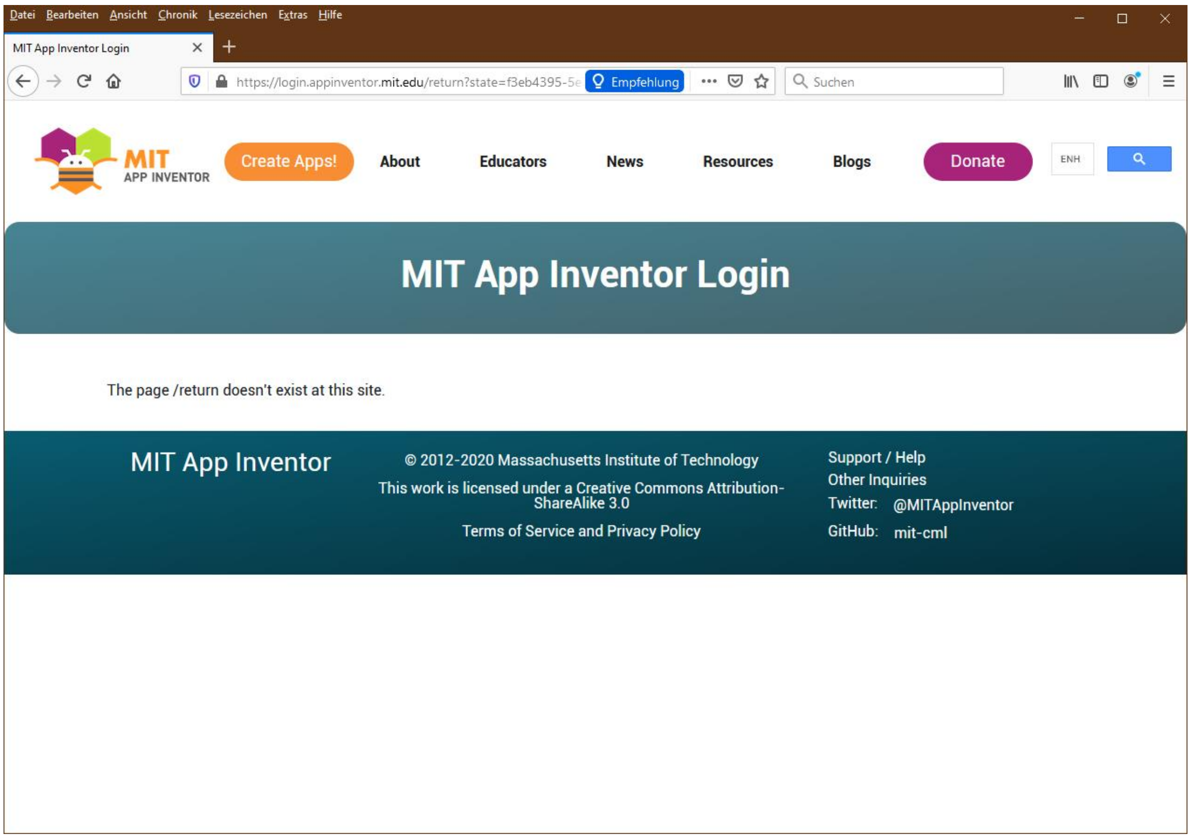
Geschlecht: Ich möchte dies nicht beantworten

Warum wir nach diesen Informationen fragen

Zurück | Weiter

Ihre personenbezogenen Daten sind bei uns sicher und geschützt

After the usual settings for privacy you will land on this page.


Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

MIT App Inventor Login

https://login.appinventor.mit.edu/return?state=f3eb4395-5e

MIT APP INVENTOR | Create Apps! | About | Educators | News | Resources | Blogs | Donate

MIT App Inventor Login

The page /return doesn't exist at this site.

MIT App Inventor

© 2012-2020 Massachusetts Institute of Technology

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0

Terms of Service and Privacy Policy

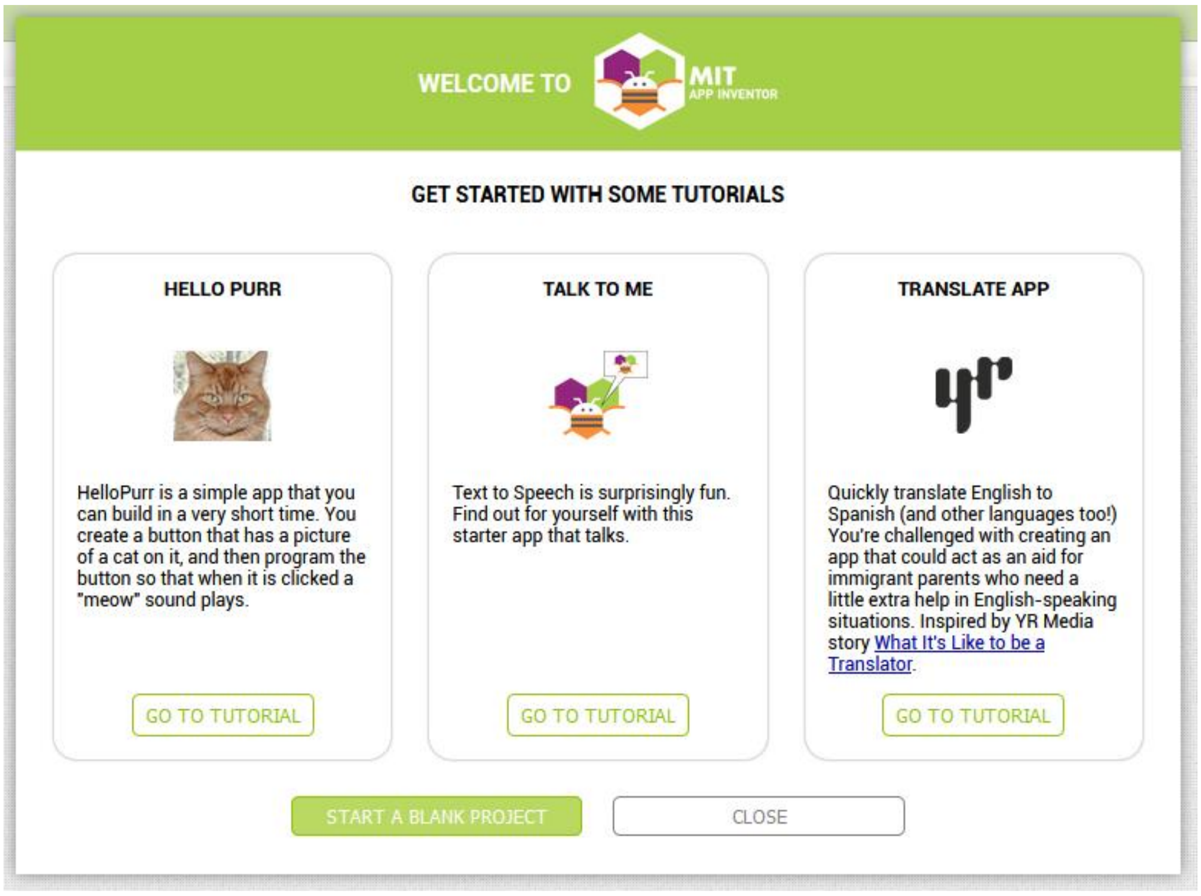
Support / Help

Other Inquiries

Twitter: @MITAppInventor

GitHub: mit-cml

The MIT APP INVENTOR link takes you back to the login. Log in with your Google account and don't be seduced by the huge range of tutorials and examples, just start a new, empty project, you want to control the Robot Car with your own app, right? Of course you do!


WELCOME TO MIT APP INVENTOR

GET STARTED WITH SOME TUTORIALS

HELLO PURR

HelloPurr is a simple app that you can build in a very short time. You create a button that has a picture of a cat on it, and then program the button so that when it is clicked a "meow" sound plays.

GO TO TUTORIAL

TALK TO ME

Text to Speech is surprisingly fun. Find out for yourself with this starter app that talks.

GO TO TUTORIAL

TRANSLATE APP

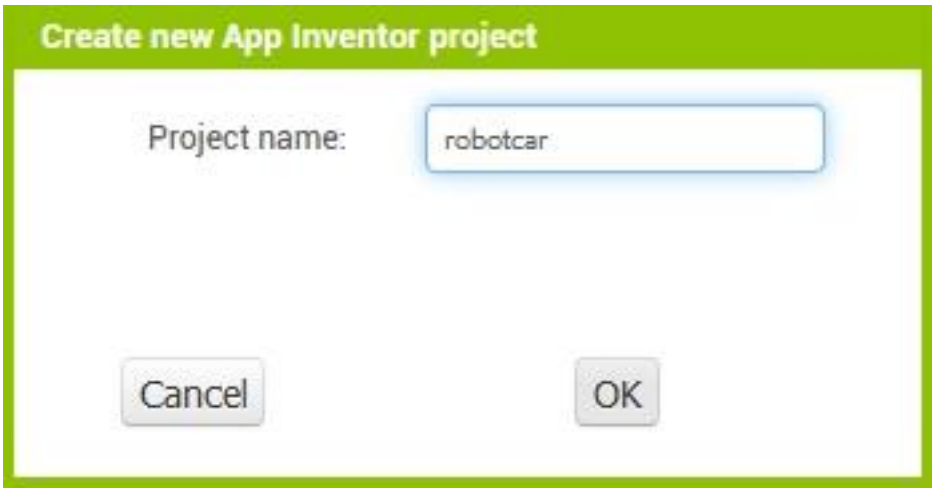
Quickly translate English to Spanish (and other languages too!) You're challenged with creating an app that could act as an aid for immigrant parents who need a little extra help in English-speaking situations. Inspired by YR Media story What It's Like to be a Translator.

GO TO TUTORIAL

START A BLANK PROJECT

CLOSE

But I also know that later on you will eagerly soak up everything that AI2 has to offer you - at least that's how I felt.


Create new App Inventor project

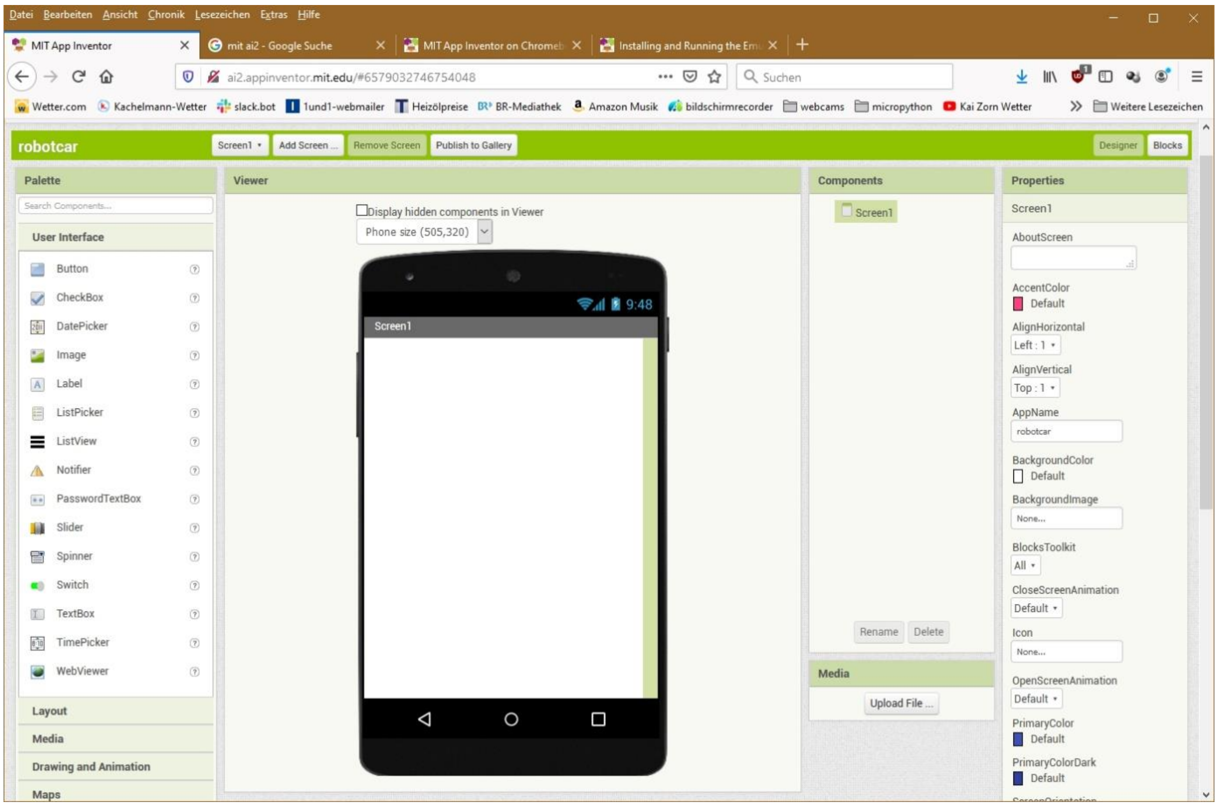
Project name: robotcar

Cancel

OK

Setup of the screen objects

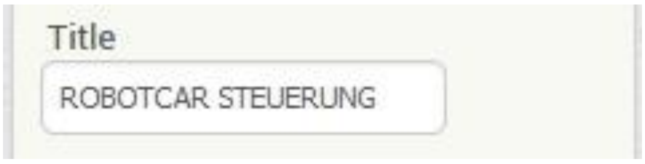
Let's take a look at the page on which the layout of the app is being designed. In the Palette section you will find the ingredients. The middle area, Viewer, is your workspace. To the right of this in the Components section, the hierarchy of the components used is shown and to the right, in Properties, the properties of the currently selected component. You can see all of this if you have selected Designer in the green line above these areas on the far right.



AppName: robotcar
AlignHorizontal: Left : 1
AlignVertical: Top : 1
AccentColor: Default
BackgroundColor: Default
BackgroundImage: None...
BlocksToolkit: All
CloseScreenAnimation: Default
Icon: None...
OpenScreenAnimation: Default
PrimaryColor: Default
PrimaryColorDark: Default

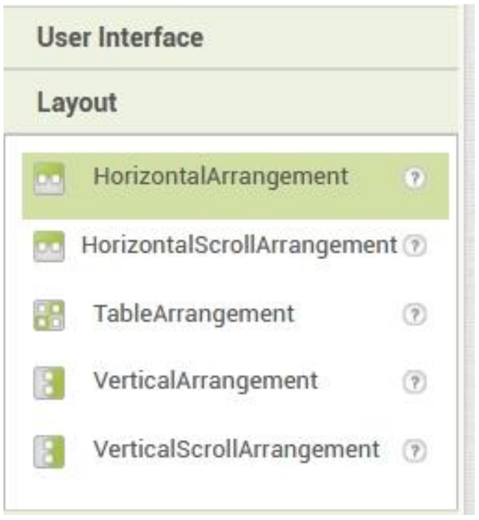
We are now going to add piece by piece to the layout and set the properties of the components. All properties that are not explicitly listed remain on the default setting.

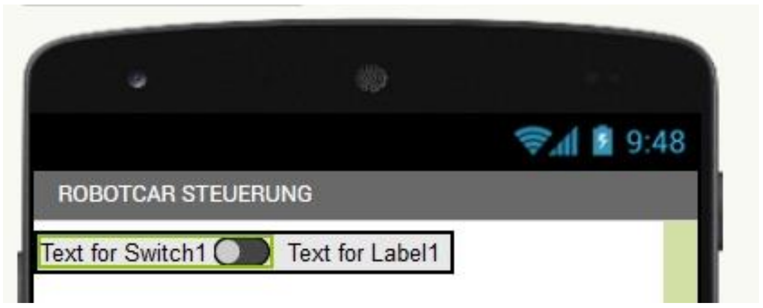
The screen component is already activated. We give it the title ROBOTCAR CONTROL. Scroll down in Properties and enter the text under Title.



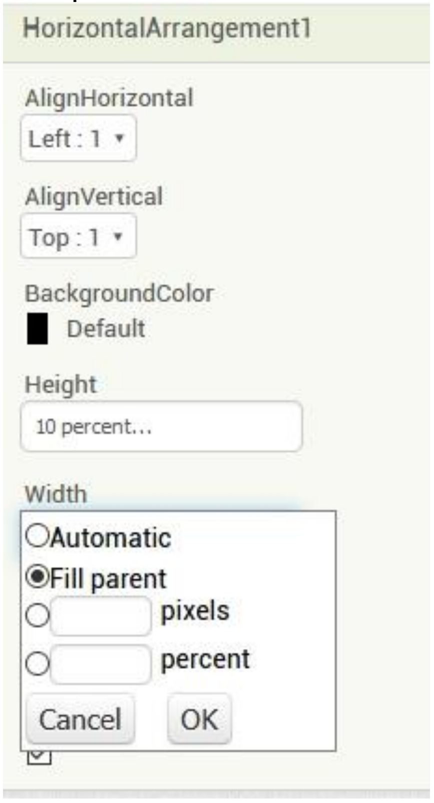
Title: ROBOTCAR STEUERUNG

A switch and a text window should appear in the first line. So get a switch and a label from the User Interface folder and drag them onto the surface of the phone. What, the label doesn't want to live next to the switch. Give both of them a tidy home, get a horizontal arrangement from the Layout folder and set it using the switch. Now you can put the Switch1 and Label1 elements in them and look, now they also sit peacefully next to each other.





Let's distribute the properties. Select the HorizontalArrangement1 entry in Components. Click the box of Width, Fill Parent and OK.


HorizontalArrangement1

AlignHorizontal
Left : 1

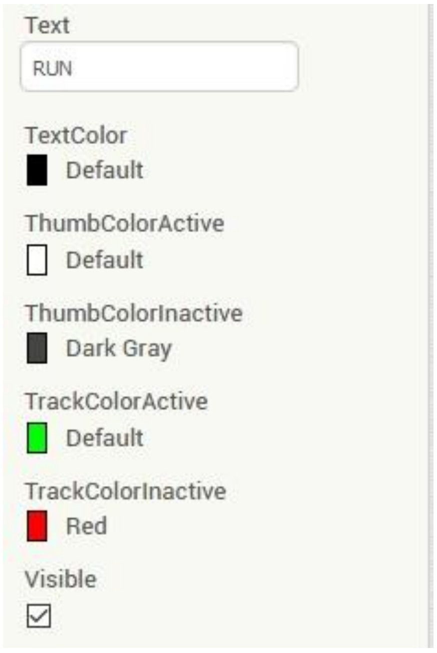
AlignVertical
Top : 1

BackgroundColor
Default

Height
10 percent...

Width
Automatic
[✓ Fill parent]
pixels
percent

In Components, select Switch1. Left click on Rename, then enter the new name MainSwitch. Change the properties as follows.



Text
RUN

TextColor
■ Default

ThumbColorActive
□ Default

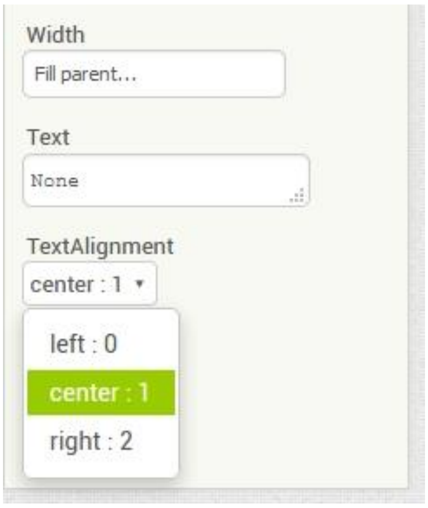
ThumbColorInactive
■ Dark Gray

TrackColorActive
🟩 Default

TrackColorInactive
🟥 Red

Visible
☑

Benennen Sie **Label1** in **MyIP** um und stellen Sie die Eigenschaften auf folgende Werte ein.



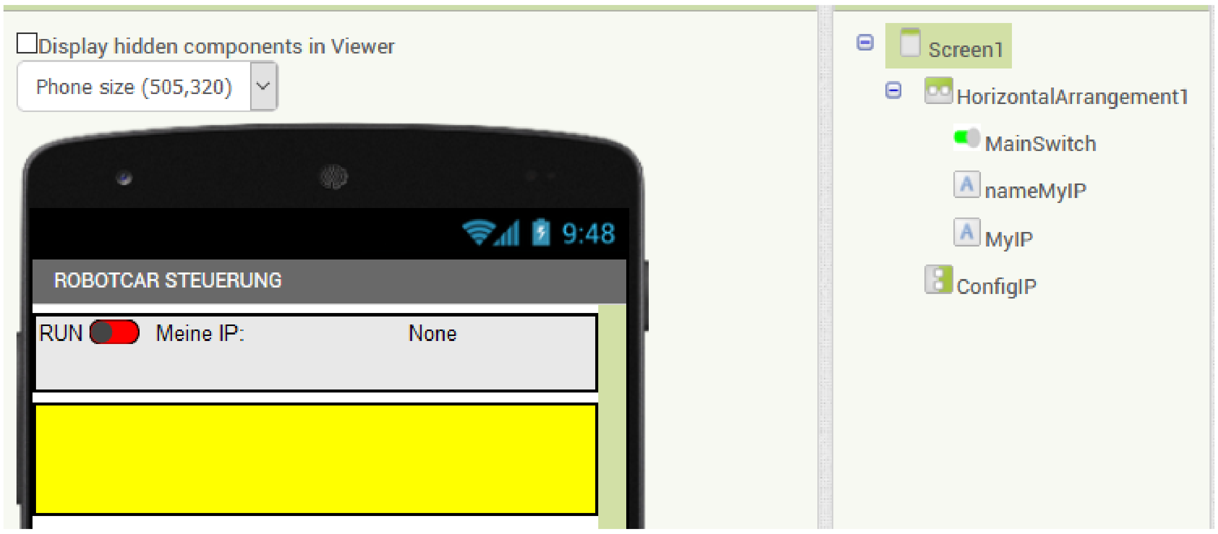
Width
Fill parent...

Text
None

TextAlignment
center : 1

Insert another label with the name nameMyIP and the text My IP between MainSwitch and MyIP.

The next arrangement is called ConfigIP, it is vertical, the background is yellow, the height is 20% of the display and it takes up the entire width of the screen. I think you can do it!?



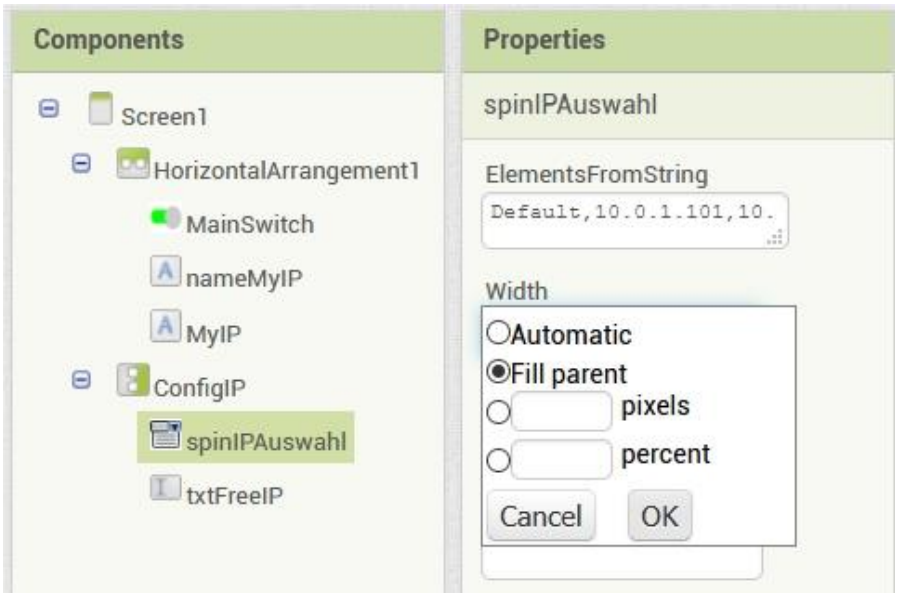
Now to the setup. A spinner is a drop-down list from which entries can be selected by touch. The entries are determined by a list. The terms are separated by commas. The first term is already marked when the screen is displayed. A text box is an input field, the content of which is accepted when you exit.

We define a list of IP addresses that represent possible goals of the project, WLAN access point, ESP32 access point and PC for test purposes. The list looks like this: Of course, you have to adapt the IP addresses to your network environment.

Default, 10.0.1.101,10.0.2.101,10.0.1.10

The selected address should be displayed in the text box at runtime in order to communicate the selection and / or to enable changes.

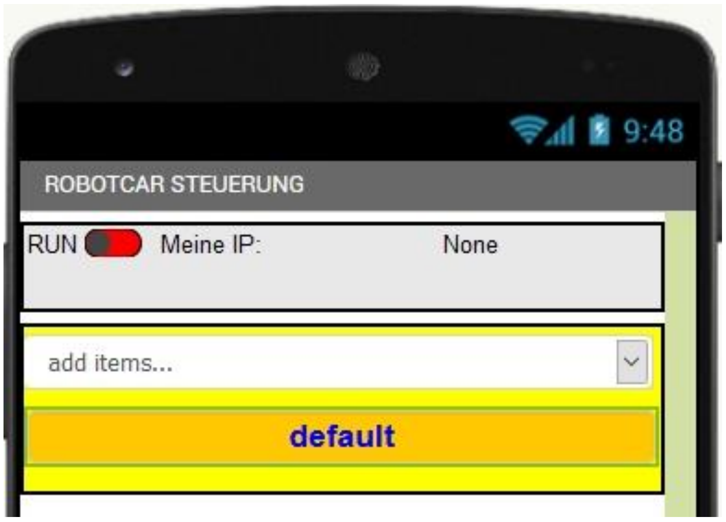
The characteristics of the spinner are shown here.



ElementsFromString
`Default,10.0.1.101,10.`

Width
Automatic [✓ Fill parent] pixels percent

Can you find out the properties of the txtFreeIP text box? I'll tell you the font size: 18.



Do you have everything? BackgroundColor: Orange, FontSize: 18, Width: Fill Parent, Text: default, TextAlignment: center 1, TextColor: Blue.

A TableArrangement now follows with 2 columns and 2 rows. I call it DriveData. It fills the entire width of the screen and contains 4 labels, the two on the left, txtVelo and txtDir, for description and the right, lblVelo and lblDir, for indicating the speed and direction of travel.

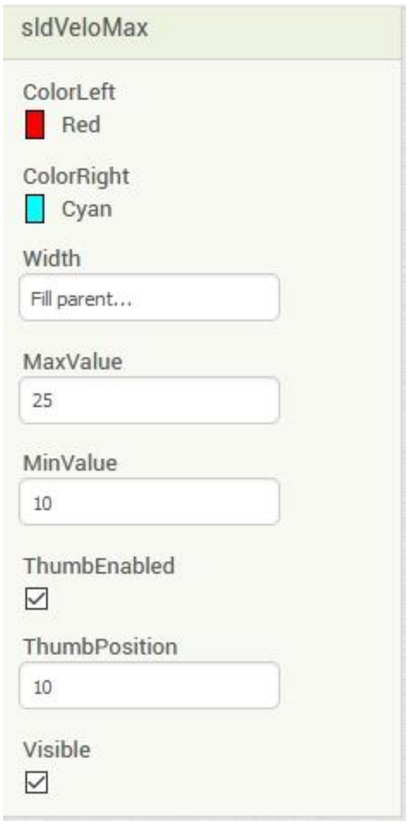
txtVelo and txtDir: FontSize: 14, Textalignment: right 2, Width: 50% Text: Speed | direction

lblVelo and lblDir: FontSize: 20, FontBold, Text: 0

In the next arrangement (horizontal, height: 20%, width: Fill Parent) there is space for two buttons, btnLight and btnMotor. The following applies to the buttons: BackgroundColor: Green, FontSize: 30, Shape: rounded, TextAlignment: center 1, Text LICHT | ENGINE.

In a subsequent horizontal arrangement, VeloMaxDisplay (Width: Fill Parent), there is a description label, txtVeloMax (Width: 50%, TextAlign: center 1, Text: Maximum speed) and a label lblVeloMax, which represents the rounded value of the following slider.

A slider (aka slider) with the name sldVeloMax with the following properties forms the end:


sldVeloMax

ColorLeft
🟥 Red

ColorRight
🟦 Cyan

Width
Fill parent...

MaxValue
25

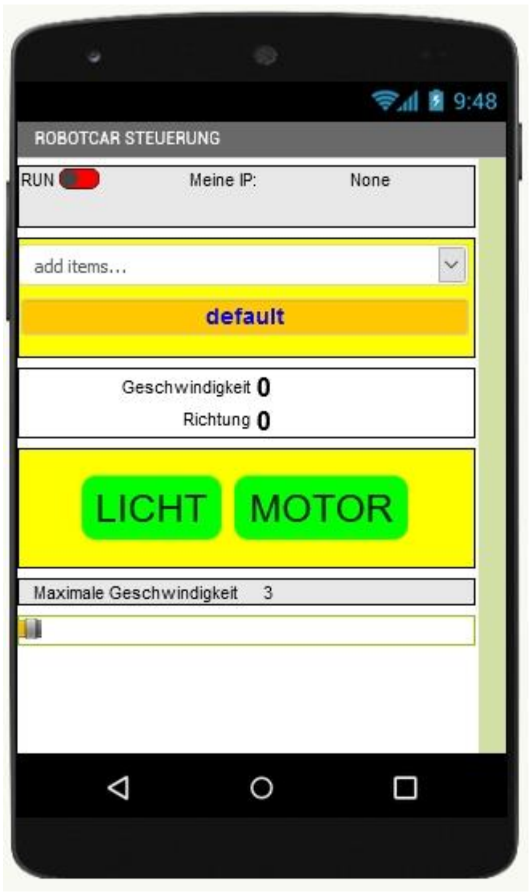
MinValue
10

ThumbEnabled
☑

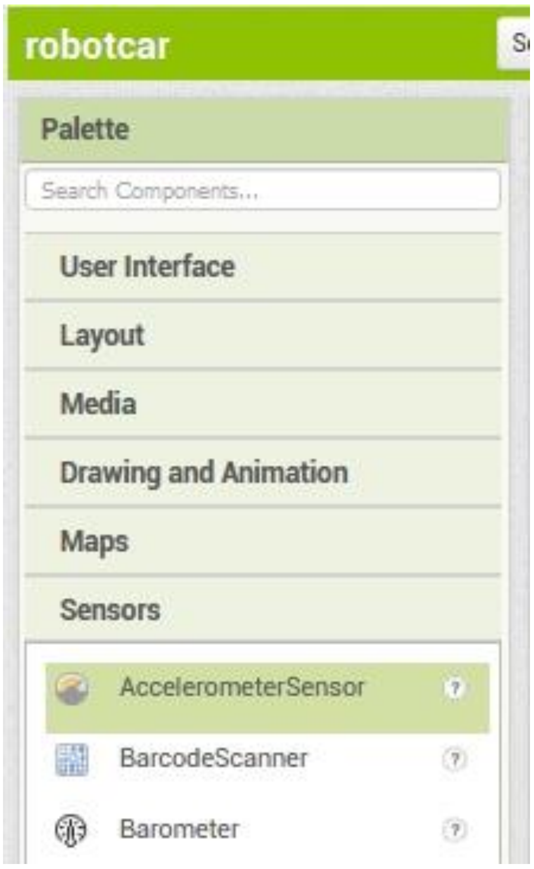
ThumbPosition
10

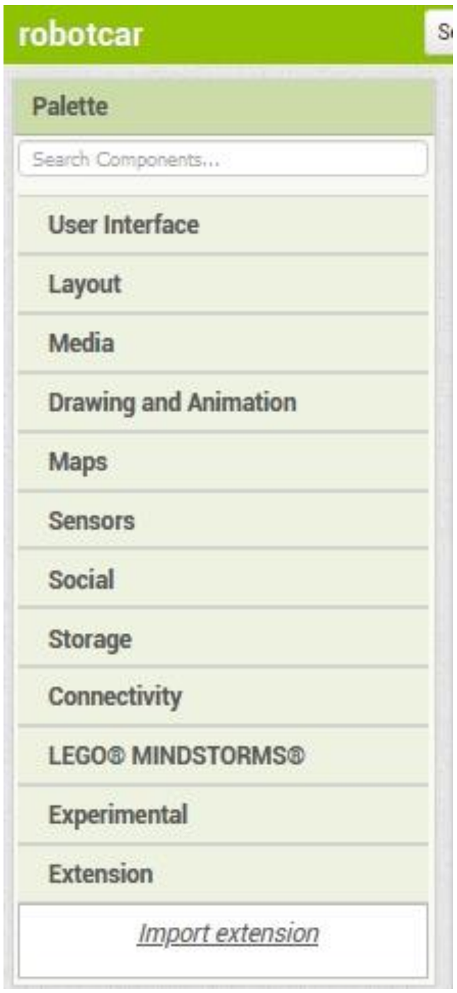
Visible
☑

Our display now looks like this.



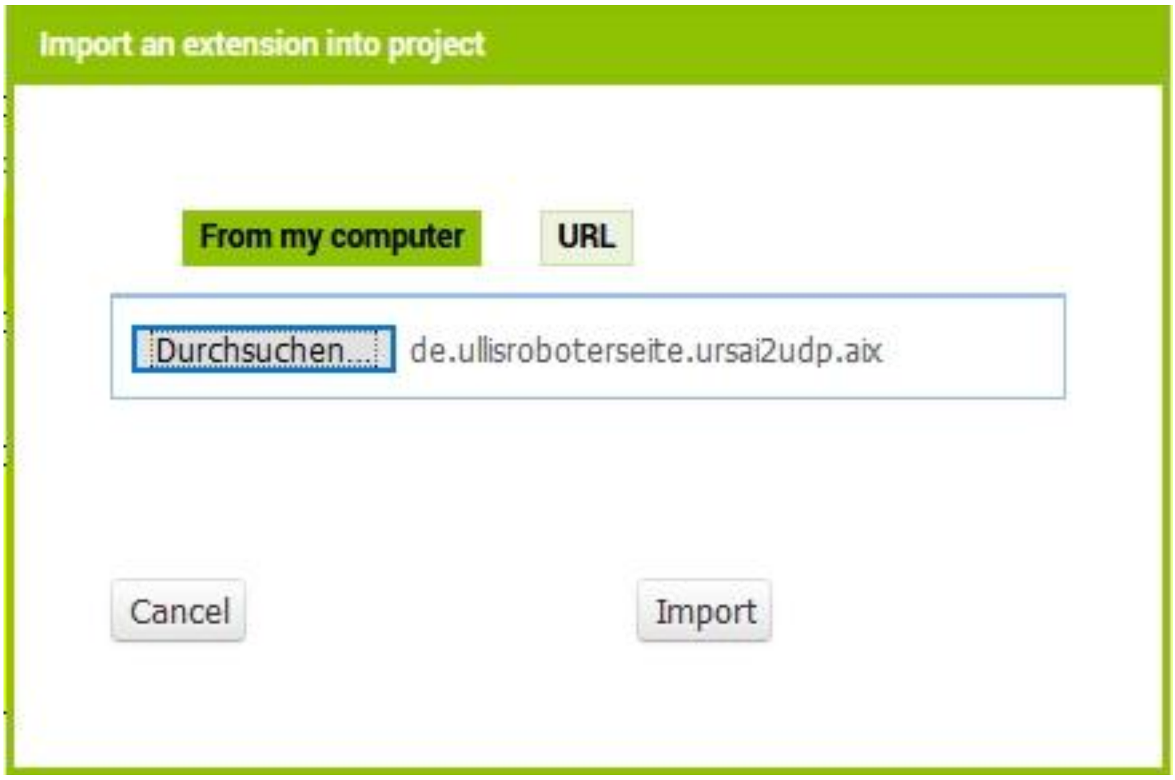
Now we are still missing two "invisible" elements, an accelerometer and the UDP transmitter. The Accelerometer Sensor can be found in the Sensors palette folder. Drag it onto the surface of the phone. It will be displayed below it.





Do you still know in which folder the de.ullisroboterseite.ursai2udp.aix file is located? We'll need this file next. We have to expand the language scope of AI2 with it. Scroll down to the bottom of the Palette, click on Extension and then on Import extension.

In the following window, after clicking on Browse, navigate to the aix file.



de.ullisroboterseite.ursai2udp.aix

After a few seconds after clicking on Import, two entries are displayed. Drag UDPXmitter onto the surface of the phone. Your mobile phone now also knows the UDP dialect and can send datagrams to the server on the Robot Car. If you want to use this feature in other projects as well, you have to repeat the import process each time after the new project has been created.

Programming

Now let's turn to programming. So far we have only provided data and objects. Now we're bringing the whole thing to life, now you can stack building blocks. In the green menu bar, use the Blocks button on the far right to switch to the programming window of AI2.

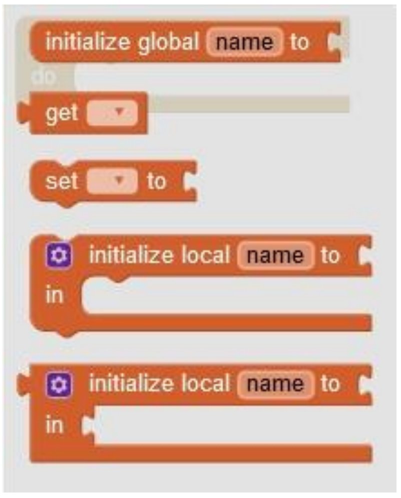
On the left is the area for program blocks or modules with built-in blocks for text, logic, data structures, mathematical functions and operators, variables and procedures. Below that you will find the blocks of the elements that we have defined so far.

In the viewer window we are now building our "houses" from the blocks for various events. You can only put together building blocks that also fit together. This corresponds to the syntax and semantic rules of MicroPython and other programming languages. You may have come across blocks like in AI2. Tools like Squeak and Scratch are based on the same method of programming. Behind this is the language Smalltalk-80.

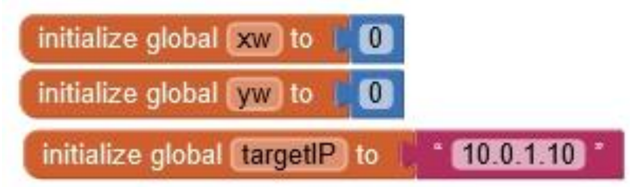
Admittedly, things are a bit bumpy for people who are used to freely writing program texts. But honestly, how often do you make a mistake and then look for the mistakes? If errors occur in Scratch or AI2, then these are logical errors, but not syntax errors. Therefore, this type of programming is well suited for beginners and is often practiced in schools.

We already prepare various variables, then we initialize the viewport. At the beginning I will describe everything very precisely. In the further course the information becomes sparse and is limited to essential things like the listing of the properties. The colors of the different areas help with orientation. Unnamed structure blocks come from the built-in area; the labeling of the named blocks reveals their origin.

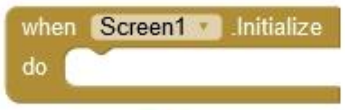
Drag an initialize global name to from Variables into the viewer. Add a 0 from Math to it. Click on name and type xw, Enter. Repeat the process for yw.

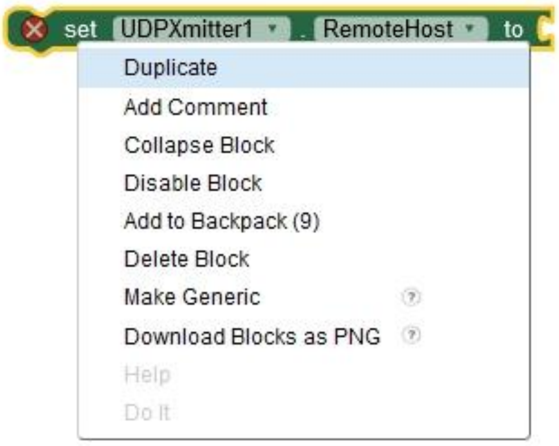


Create a global variable targetIP and add a "" to it from the text. Then enter the IP address of your PC. That should be enough to demonstrate the insertion and gluing of blocks. We'll add more variables later.


initialize global xw to 0
initialize global yw to 0
initialize global targetIP to " 10.0.1.10 "

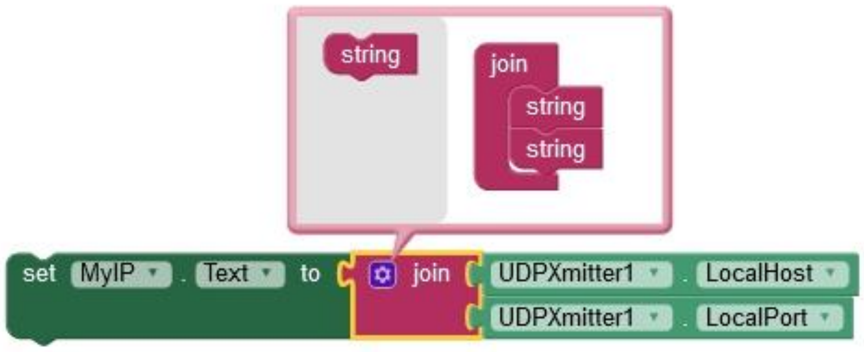
Now pull a when Screen1 initialize bracket from Screen1.

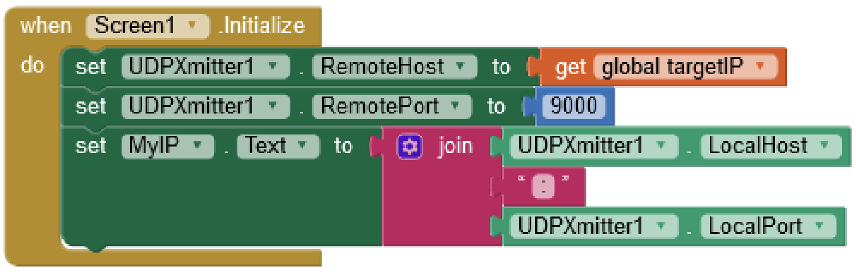




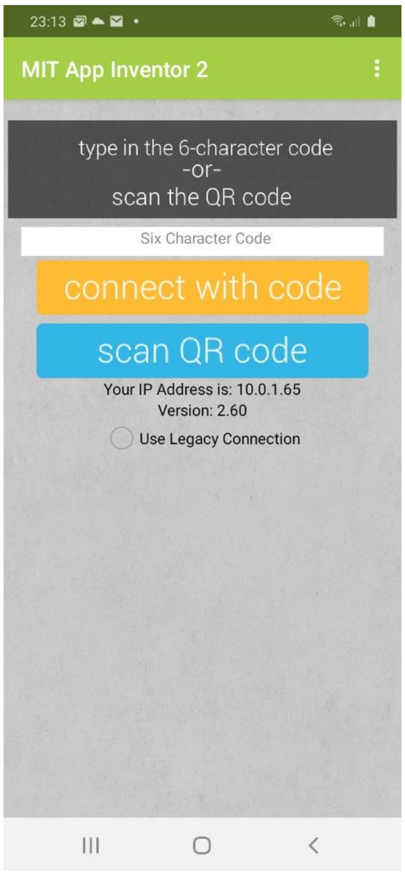
A right click on the set UDPXmitter1 strip from UDPXMitter1 and a click on Duplicate create a duplicate of the block. Glue the second to the first and in the second block, change RemoteHost to RemotePort. Get a get from Variables and a 0 from Math. Glue the get to RemoteHost and the 0 to RemotePort. Change the 0 to 9000 and in the get block select targetIP.

Get a set MyIP.Text to from MyIP and stick a join of text to it. Glue a LocalHost and LocalPort from UDPXmitter1 to the join. You need another point of contact for the separating colon. A click on the blue symbol opens the setting window. Pull string from the left half between the two in the right half and then add the colon.

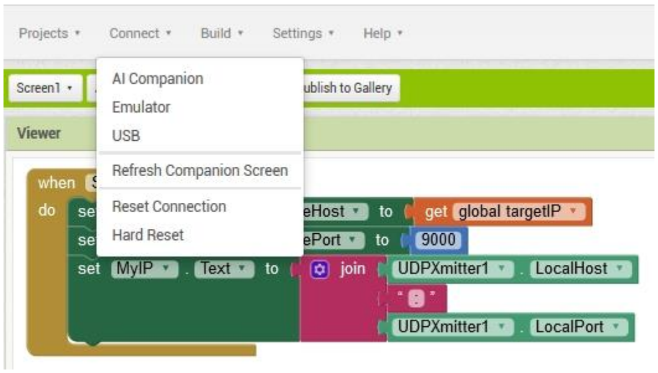



when Screen1 .Initialize
do set UDPXmitter1 . RemoteHost to get global targetIP
set UDPXmitter1 . RemotePort to 9000
set MyIP . Text to join UDPXmitter1 . LocalHost
" : "
UDPXmitter1 . LocalPort

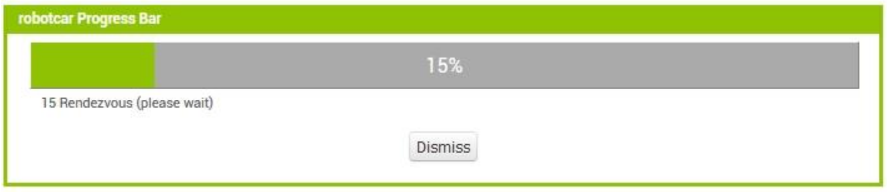
If you don't have any error messages and warnings, you will surely be curious to see how it all looks on your phone. Connect the mobile phone to the WLAN router and start the AI2 Companion on the mobile phone.



It is particularly easy to establish contact between mobile phone and PC using the QR code scan. On the PC side, start the process by clicking on Connect and AI Companion.

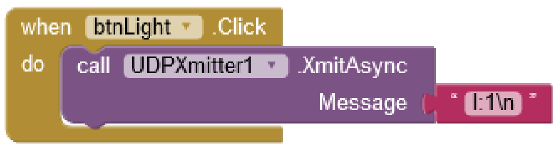


Now scan the QR code with your mobile phone.

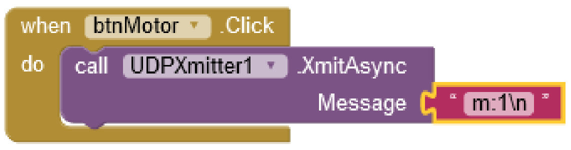


The progress bar runs through and then you have your draft with the first programming result on the mobile phone display. The local IP of the smartphone is displayed to the right of My IP. You can also tap the switch, open the spinner list and operate the slider. So that this also triggers actions, we have to add a lot of code. By the way, don't be alarmed, the connection to the cell phone is cut again relatively quickly if no action is taken. This will not delete or change anything, you just have to reconnect. If you later install the entire app on the mobile phone, there is no longer any automatic disconnection.

This is how you bring the buttons for the driving lights and the motor relay to life.



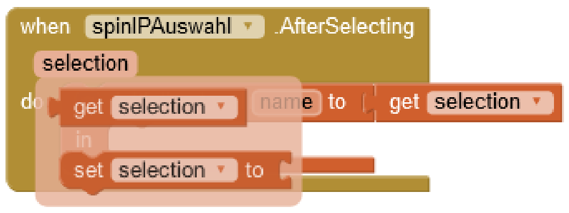
```
when btnLight .Click
do  call UDPXmitter1 .XmitAsync
                         Message " l:1\n "
```



```
when btnMotor .Click
do  call UDPXmitter1 .XmitAsync
                         Message " m:1\n "
```

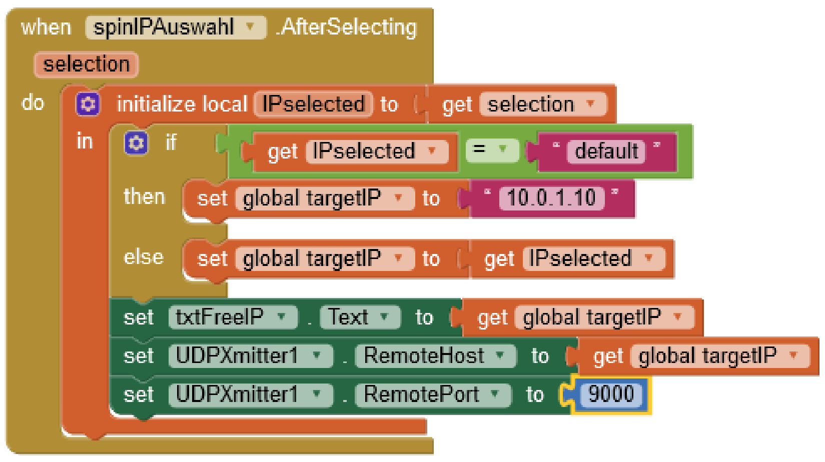
As the names of the when brackets show, they come from the block pool of the buttons btnLight and btnMotor. the function calls come from UDPXmitter1. The command strings must be terminated with a linefeed, that is done by "\ n".

Now it’s up to the weirdo. To handle the selection, we create a local variable IPselect and assign get selection from the selection field.



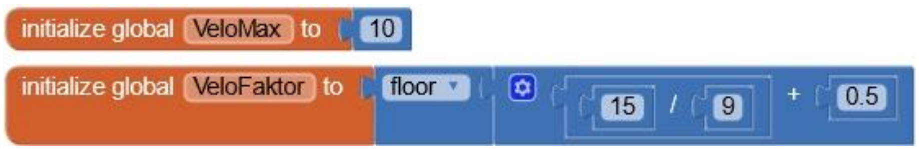
```
when spinIPAuswahl .AfterSelecting
  selection
do  get selection    name to  get selection
    in
    set selection to
```

The rest of the "filling" is self-explanatory.

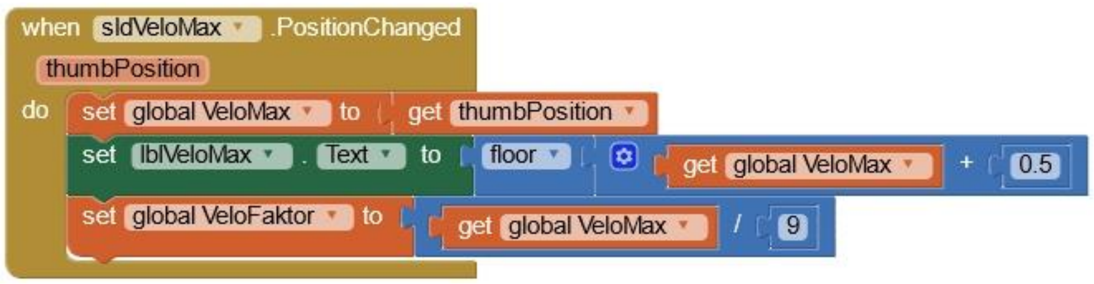


```
when spinIPAuswahl .AfterSelecting
  selection
do  initialize local IPselected to  get selection
    in  if     get IPselected  =  " default "
        then   set global targetIP to  " 10.0.1.10 "
        else   set global targetIP to  get IPselected
        set txtFreeIP . Text to  get global targetIP
        set UDPXmitter1 . RemoteHost to  get global targetIP
        set UDPXmitter1 . RemotePort to  9000
```

To determine the speed steps, we need a function that converts the floating raw values from the accelerometer into usable integer values in the permissible range. The slider is used to set the upper limit of the range. So that you can see the set value, I assign the rounded value of the slider to lblVeloMax.Text. I need two new global variables for the calculations.



```
initialize global VeloMax to 10
initialize global VeloFaktor to floor ( 15 / 9 ) + 0.5
```



```
when sldVeloMax .PositionChanged
  thumbPosition
do  set global VeloMax to get thumbPosition
    set lblVeloMax . Text to floor get global VeloMax + 0.5
    set global VeloFaktor to get global VeloMax / 9
```

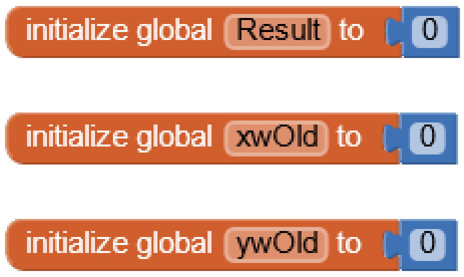
To calculate the speed steps, I need the ratio of the maximum possible speed step and the maximum absolute values of the accelerometer chip. I measured 9 of these in all directions. The calculated value is available in VeloFaktor.

Just as the raw values come from the accelerometer, the signs for both axes are complementary. In the function procedure2, after the introduction of some local variables, the signum vz of the parameter x is calculated and, if not equal to zero, inverted.

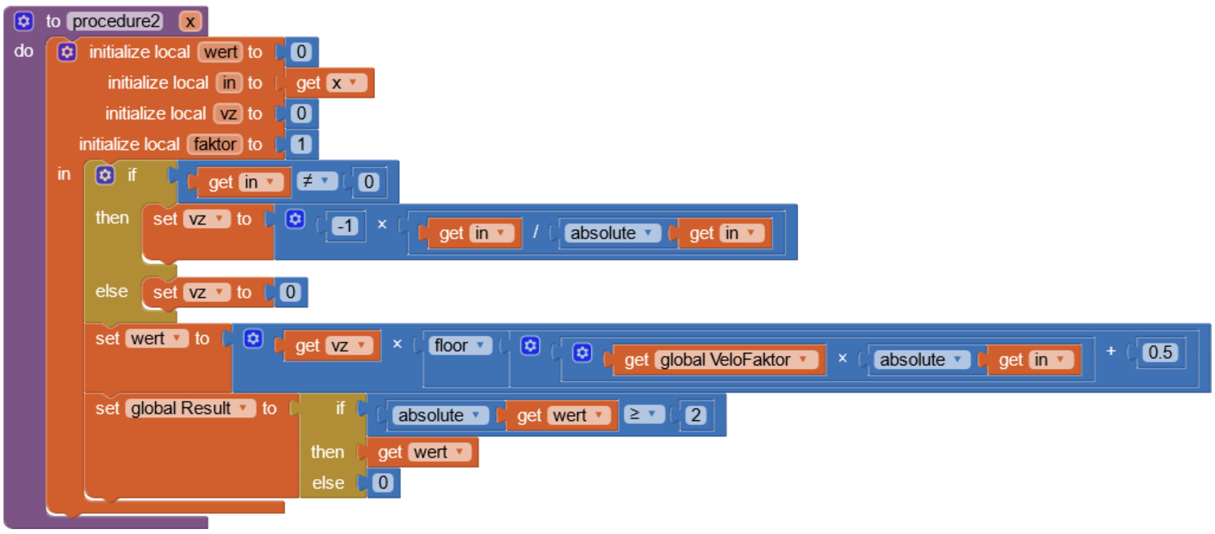
The speed step value is calculated using the following formula:

wert =vz*int((VeloFaktor * abs(x))+0,5)

The new global variable Result is used to return the result. So that this can be done in one go, we also define two flags for xw and yw from the previous run.

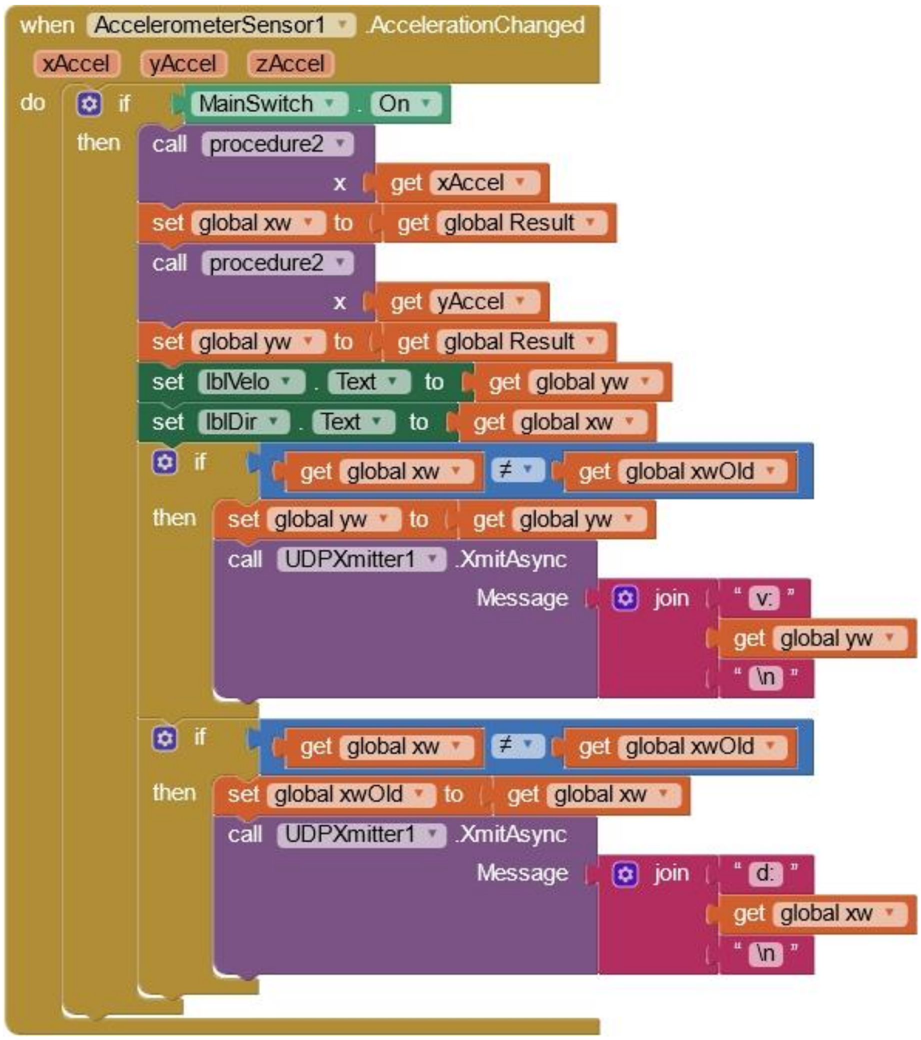


```
initialize global Result to 0

initialize global xwOld to 0

initialize global ywOld to 0
```



```
to procedure2 x
do  initialize local wert to 0
    initialize local in to get x
    initialize local vz to 0
    initialize local faktor to 1
    in  if get in ≠ 0
        then set vz to -1 × get in / absolute get in
        else set vz to 0
        set wert to get vz × floor get global VeloFaktor × absolute get in + 0.5
        set global Result to  if absolute get wert ≥ 2
                              then get wert
                              else 0
```

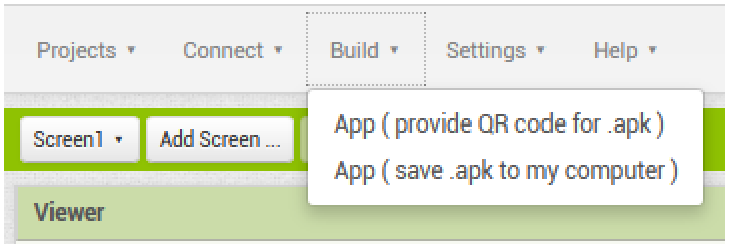
This brings us to the last block. The accelerator values are recorded here, converted to speed levels and if a value has changed, it is sent to the Robot Car. For all of this to really happen, the MainSwitch must of course be on.



```
when AccelerometerSensor1 .AccelerationChanged
  xAccel  yAccel  zAccel
do  if MainSwitch . On
    then call procedure2
                    x get xAccel
         set global xw to get global Result
         call procedure2
                    x get yAccel
         set global yw to get global Result
         set lblVelo . Text to get global yw
         set lblDir . Text to get global xw
         if get global xw ≠ get global xwOld
         then set global yw to get global yw
              call UDPXmitter1 .XmitAsync
                         Message join " v: "
                                      get global yw
                                      " \n "
         if get global xw ≠ get global xwOld
         then set global xwOld to get global xw
              call UDPXmitter1 .XmitAsync
                         Message join " d: "
                                      get global xw
                                      " \n "
```

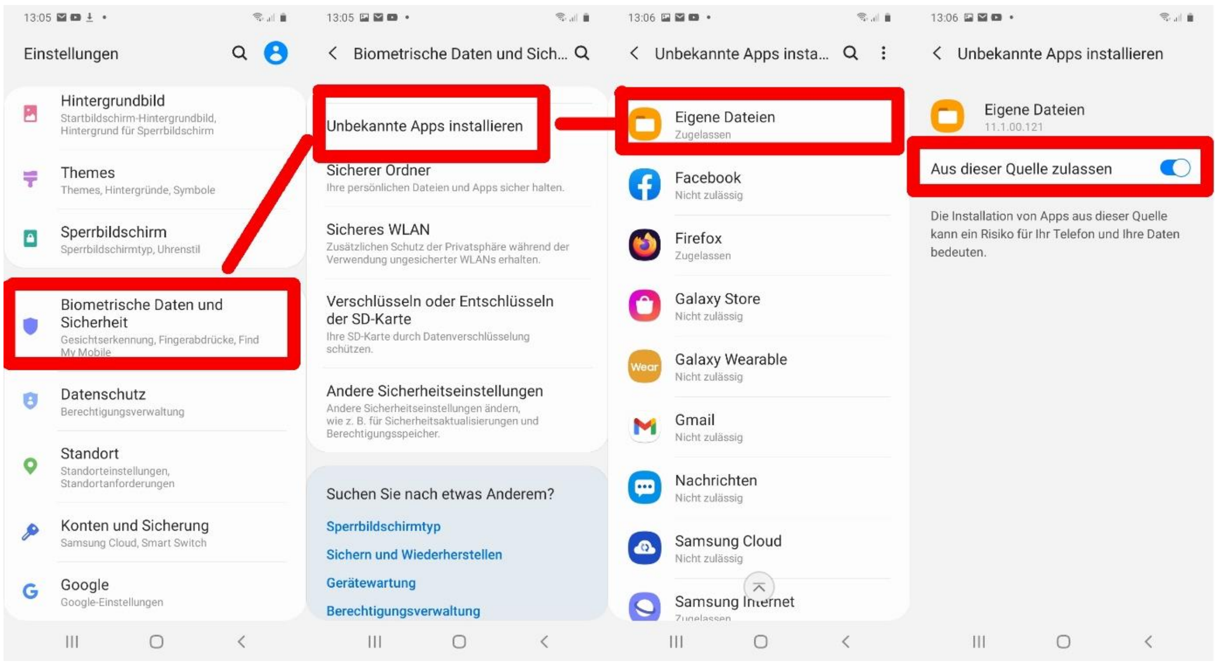
Now everything is in the box and it is time to complete the final function test on the mobile phone. Start the Robot Car and connect your mobile phone and PC as you did above. Mainswitch on, tilt the mobile phone forward and off you go. Take a few bends, make a U-turn - perfect!

Then let's turn the project draft into a permanent app that you can install on the phone. The application is always compiled in the AI cloud. Your project will also be saved there and the apk file will be downloaded from there. The Build button offers two options. With the first option, a QR code will be displayed approx. 1 to 2 minutes after clicking. It contains the download url. Scan it in. The installation starts after the download. Confirm the security questions all with 'yes'



The second option offers the download to your PC after the translation. Save the apk file in any directory and then transfer it to your mobile phone using Bluetooth or another method (e-mail, WWW, ...). robotcar.apk (apk = Android PacKage) probably ends up in the download folder and it is in 'My Documents'. First check whether setups can be carried out from this folder.

settings …



Einstellungen

Hintergrundbild
Startbildschirm-Hintergrundbild, Hintergrund für Sperrbildschirm

Themes
Themes, Hintergründe, Symbole

Sperrbildschirm
Sperrbildschirmtyp, Uhrenstil

Biometrische Daten und Sicherheit
Gesichtserkennung, Fingerabdrücke, Find My Mobile

Datenschutz
Berechtigungsverwaltung

Standort
Standorteinstellungen, Standortanforderungen

Konten und Sicherung
Samsung Cloud, Smart Switch

Google
Google-Einstellungen

Biometrische Daten und Sich...

Unbekannte Apps installieren

Sicherer Ordner
Ihre persönlichen Dateien und Apps sicher halten.

Sicheres WLAN
Zusätzlichen Schutz der Privatsphäre während der Verwendung ungesicherter WLANs erhalten.

Verschlüsseln oder Entschlüsseln der SD-Karte
Ihre SD-Karte durch Datenverschlüsselung schützen.

Andere Sicherheitseinstellungen
Andere Sicherheitseinstellungen ändern, wie z. B. für Sicherheitsaktualisierungen und Berechtigungsspeicher.

Suchen Sie nach etwas Anderem?
Sperrbildschirmtyp
Sichern und Wiederherstellen
Gerätewartung
Berechtigungsverwaltung

Unbekannte Apps insta...

Eigene Dateien
Zugelassen

Facebook
Nicht zulässig

Firefox
Zugelassen

Galaxy Store
Nicht zulässig

Galaxy Wearable
Nicht zulässig

Gmail
Nicht zulässig

Nachrichten
Nicht zulässig

Samsung Cloud
Nicht zulässig

Samsung Internet
Zugelassen

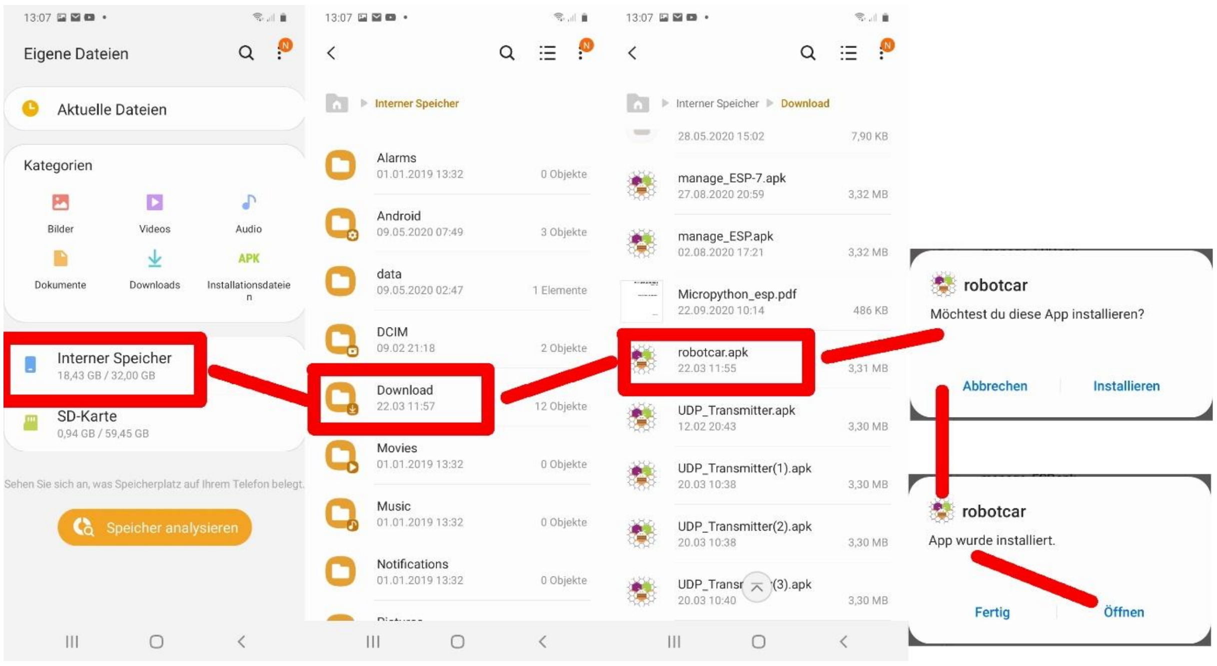
Unbekannte Apps installieren

Eigene Dateien
11.1.00.121

Aus dieser Quelle zulassen

Die Installation von Apps aus dieser Quelle kann ein Risiko für Ihr Telefon und Ihre Daten bedeuten.

Own files …


Eigene Dateien
Aktuelle Dateien
Kategorien
Bilder
Videos
Audio
Dokumente
Downloads
Installationsdateien
Interner Speicher
18,43 GB / 32,00 GB
SD-Karte
0,94 GB / 59,45 GB
Sehen Sie sich an, was Speicherplatz auf Ihrem Telefon belegt.
Speicher analysieren
Interner Speicher
Alarms 01.01.2019 13:32 0 Objekte
Android 09.05.2020 07:49 3 Objekte
data 09.05.2020 02:47 1 Elemente
DCIM 09.02 21:18 2 Objekte
Download 22.03 11:57 12 Objekte
Movies 01.01.2019 13:32 0 Objekte
Music 01.01.2019 13:32 0 Objekte
Notifications 01.01.2019 13:32 0 Objekte
Interner Speicher ▸ Download
28.05.2020 15:02 7,90 KB
manage_ESP-7.apk 27.08.2020 20:59 3,32 MB
manage_ESP.apk 02.08.2020 17:21 3,32 MB
Micropython_esp.pdf 22.09.2020 10:14 486 KB
robotcar.apk 22.03 11:55 3,31 MB
UDP_Transmitter.apk 12.02 20:43 3,30 MB
UDP_Transmitter(1).apk 20.03 10:38 3,30 MB
UDP_Transmitter(2).apk 20.03 10:38 3,30 MB
UDP_Transmitter(3).apk 20.03 10:40 3,30 MB
robotcar
Möchtest du diese App installieren?
Abbrechen
Installieren
robotcar
App wurde installiert.
Fertig
Öffnen

After the final test, you should revoke the permission to install apps from the folder 'My Documents' for security reasons.

You can download an overview of the program blocks as a PDF. The robotcar.aia project file is also available for download, as is the robotcar.apk setup file. Unfortunately there is no listing for this project.

I hope you enjoy further experimenting with the Micropython and the Robot Car. You can expand the mobile phone control with the release of the start for the autonomous driving of your e-car. Of course, routines have to be written for this, for example for following a track line. The same can be done there by adding another button to the ESP32 hand control. Or how about an automatic garage door control via infrared modules when the car approaches? The ESP32 offers all the prerequisites for this.

Catch the challenge!

Linkliste:

Teil1	HTML-Format
Teil2	HTML-Format
Teil1	deutsches PDF
	englisches PDF
Teil2	deutsches PDF
	englisches PDF
Teil3	deutsches PDF
	englisches PDF